

DISPLAYING WEB GRAPHICS

Controlling the Appearance of Graphics in Web Pages

In this chapter, you will:

- ◆ Prepare images for display on different hardware, browsers, and operating systems
- ◆ Learn how colors are represented on the Web
- ◆ Control the appearance of Web graphics with HTML
- ◆ Avoid common mistakes when using images in HTML

When creating any media, you need to understand the environment in which your work will be seen. Users will view your work on different operating systems, including Windows and Macintosh. They will use different browsers, and even different versions of Internet Explorer and Netscape Navigator. The size of their monitors also will vary. These different elements make up the user environment.

You cannot create graphics that will appear exactly the same way in all environments, so you need to design for the majority, yet include as many other environments as possible. On the Web, most people use Microsoft Windows and Internet Explorer. As you create Web graphics, you need to make sure that those users see the images as you intend, and still ensure that Mac and Netscape users are not left behind. This chapter explains how to prepare images for the widest range of hardware, browsers, and operating systems.

One way that environments differ is in the way browsers represent colors. An important part of designing graphics and pages for all browsers is having an understanding of how color is defined in HTML. Web colors are always defined in terms of RGB color, but you use special syntax to do so. This chapter also explains how colors are represented on the Web.

To create Web pages that display your graphics, you can code the HTML in a text file or use an editor such as Dreamweaver. Most designers prefer to use an HTML editor because the interface simplifies the process of producing

Web pages. However, HTML editors usually do not lay out pages with the accuracy you need. You should know at least some HTML so you can fix the code generated by the editors. This chapter discusses how you can control the appearance of Web graphics with HTML, and avoid the common mistakes in using images in HTML.

PREPARING CROSS-PLATFORM IMAGES

Web users have different kinds of hardware for viewing Web pages, and they use different browsers running on different operating systems. Ideally, you will create cross-platform pages and images, meaning they look similar, regardless of the hardware, browsers, or operating systems used. Creating cross-platform images often is impossible because environments differ dramatically. Someone using Netscape 3 on a Macintosh with a 21" screen sees pages and graphics differently from someone using Internet Explorer 5 on a Windows computer with a 13" monitor. Your goal is to develop images that look the way you want on the majority of systems, and look close enough on most others.

Considering Hardware

The hardware that most affects how others view your images is the standard desktop monitor. This is the tool most people use to surf the Web and view Web graphics. However, many people print pages before reading them, so you also need to be aware of how your graphics appear when printed. Although WebTV® is still in its infancy, television and the Internet are expected to be integrated in the future, making the television another display device to consider.

Computer screens, printers, and televisions all have different color **gamuts**. The gamut of display hardware is the range of colors that the hardware produces. In general, printers have larger gamuts than monitors and televisions. They can produce a broader range of saturated colors using ink on paper than the other hardware can using light on a screen. Televisions have the smallest gamuts. The colors that they can display are a subset of the colors that computer monitors can display.

The following sections discuss considerations for designing Web graphics for a variety of monitors, televisions, and printers.

Considering Monitors

A standard desktop monitor uses cathode-ray tubes (CRTs) to project light against the back of the glass screen; these monitors are called CRT monitors and work by emitting light through the screen. The other type of monitor uses a liquid crystal display (LCD) similar to that used in calculators. LCD screens can be much thinner than CRT monitors and work by selectively blocking light. They do not emit as much light as CRT monitors and are more difficult to view from an angle. Monitors differ from each other

in size, resolution, and color depth. The images you create can have a certain size and color palette on one monitor and a different size and colors on another.

Designing for Monitor Size The size of a desktop monitor is measured diagonally on the glass part of the screen. Some manufacturers embellish the size of their monitors by measuring from corner to corner of the monitor, including the plastic case. You can find the true size by measuring the visible area of the screen with a ruler or tape measure. Sizes can range from 9 to 21" or more, but most people have 13-, 15-, or 17" screens. The size of your users' monitors affects your decisions about page layout more than it affects your decisions about graphics production. As a Web developer and producer, you need to make sure your pages fit within the viewing screen by limiting the width of Web pages. Some users set their browsers to fill the screen, and others use the default width, which covers only about half of the screen area. If the Web pages you create are no wider than about 600 pixels, they will fit on most browsers on most monitors.

Graphic artists also need to work within parameters. If you have a very large monitor, such as 21" or more, you might be accustomed to making very large images. An image that is 850 pixels wide may look good on your 21" monitor, but will not even fit on many 13" monitors, which usually can accommodate only 640 or 800 pixels, depending on resolution. When your images are placed on a Web page, you should preview the page in a mid-size monitor (13–17") to see how the average user will see it.

Designing for Monitor Resolution Most monitors have resolutions of 72 pixels per inch (ppi), and an image that is 144 pixels high appears 2" high. Some monitors, especially those that are sold with high-end graphics workstations, have resolutions of 96 ppi. On these monitors, the same 144-pixel-high image appears only 1½" high. The majority of users have 72 ppi monitors, so if you design on a standard monitor, you can be confident that most users see images the same size that you do.

Designing for Monitor Color The color displayed by a monitor is not determined entirely by the monitor, but also by the graphics card and the RAM in the computer itself. Depending on the card and the amount of memory, a monitor might display only 8-bit color (256 colors), 16-bit color (thousands of colors), or 24-bit color (millions of colors). The average user has a 16-bit monitor. Because of the difference in monitor color, an image can have smooth gradients when viewed on a 24-bit color monitor, but show slight banding or dithering when viewed on a 16-bit monitor. If you design on a 24-bit monitor, users with 16-bit monitors might see color artifacts that are invisible to you. You should preview your graphics on 16-bit and 8-bit monitors to see how they appear to users with those monitors.

The difference between monitor color depths is the same as between 8-bit and 24-bit images in an image editor. Lower color depth means banding or dithering in the image. The difference between 16-bit and 24-bit is usually not noticeable except for high-color images with smooth color gradients. Figure 3-1 shows a simulation of how a high-color image appears on monitors with three different color depths.

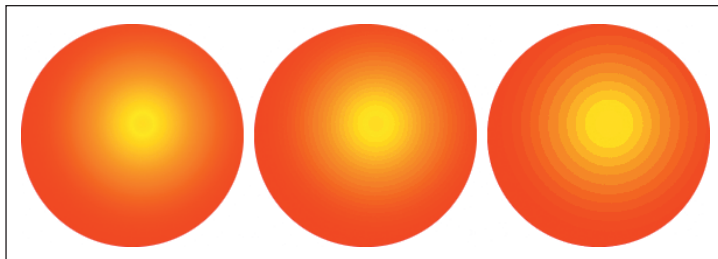


Figure 3-1 A high-color image displayed at different color depths

As a designer, you also might be accustomed to using high-color monitors and never see what your images look like on a 16-bit system. If you work on a 24-bit monitor, occasionally set your monitor to a lower color depth to see how your images appear to other users.

Both CRT and LCD monitors display colors that are less rich than in real life. This is especially true for bright shades of green. Limitations in the phosphors in computer monitors means they cannot re-create the green of objects such as the leaves of a healthy plant. To compensate for this, many graphics use a darker shade of most colors. Although a dark shade actually has less color, it appears richer to our eyes.

The colors at the top of Figure 3-2 are fully saturated. The colors on the bottom actually have less saturated color but appear to have more.

Designing for LCD Monitors LCD screens are used in devices such as cell phones and handheld computers, where their thinness is an advantage. These devices often display fewer colors because of memory limitations. Some can display up to 8-bit (256) color, but most can display only 1-bit color in black and white. LCD screens also are used for laptops and in more expensive, flat-panel desktop monitors. LCD panels typically have crisper lines than CRT monitors and less glare.

Most users will view your graphics on CRT or LCD screens, at a resolution and color depth equal to or less than what you use. Design your Web graphics to suit your users' monitors, which probably have smaller screen size and lower color depth than what you use.



Figure 3-2 Compensating for weak saturation with darker colors

Designing for a TV Screen

WebTV® from WebTV Networks Inc. is a commercial product that allows you to surf the Web using a television instead of a computer monitor. Even though televisions also use CRTs to display images, they display images differently from monitors, primarily in their resolution and use of color.

Design for television output only if you are certain you need to cater to WebTV users. Currently, only users of WebTV use television as a display device for Web pages and graphics. You can find out how many WebTV users you have by analyzing your Web server log file. Every Web site has a server log, which records who visits the site and what browser and system they are using. You cannot use the logs to tell what people do with your Web pages, only what software they use to access and view them. For most sites, well under 1% of visits are from WebTV users. This percentage might increase as high-definition digital TV becomes more popular, and the Internet and television become more integrated.

When you design Web pages you know will be viewed on a TV monitor, consider the lower resolution, reduced color, and type of output standards the TV uses. For most pages, however, you can safely ignore these restrictions.

Designing for Television Resolution The resolution of a conventional television is much lower than that of a computer monitor. If you display a high-resolution image on a television, you see a shimmering effect, which is the result of **aliasing**: outputting high-frequency information through a lower-frequency device. Aliasing can happen in any medium when high-resolution data is output at a lower resolution. You've heard aliasing if you've ever listened to digital audio played back at the wrong rate. You've seen aliasing on TV if you've ever noticed a newscaster's striped shirt creating vibrant patterns on the screen. The stripes on the shirt have a higher resolution (lines per inch) than the lines on the TV. The TV tries, and fails, to reproduce the lines; the result is the shimmering effect.

The solution for this is to anti-alias any fine lines—lines thinner than two or three pixels. Anti-aliasing here simply means blurring. To anti-alias graphics that appear on a TV screen, use the Photoshop Gaussian Blur filter to blur the whole image at 0.5 pixels. The

image looks fuzzy on the computer screen, but looks just right on TV. You can practice using the Gaussian Blur filter in Project 3-1 at the end of this chapter.

Designing for Television Color Colors on a television screen are much less saturated than colors on monitors. Most saturated colors—especially red, but also yellow and green—are too intense for a television screen. They distort and appear to bleed, causing unwanted effects. To reduce the saturation of colors for television screens, use the NTSC Colors filter in Photoshop.

To reduce the saturation of colors for television screens:

1. In Photoshop, open the **image** you want to modify.
2. Click **Filters** on the menu bar.
3. Point to **Video**, and then click **NTSC Colors**.

The colors seem muddy when you view the image on your computer screen, but they will look fine on a TV.

Designing for NTSC Standards In the United States, parameters for output on a television must meet National Television Standards Committee (NTSC) standards. NTSC is also known as the American system. Most other countries use the Phase Alternation Line (PAL) system or the Sequential Couleur Avec Memoir (SECAM) system. NTSC standards were instituted in the 1950s. NTSC uses 60 fields per second and 525 lines per field. The other two systems are improvements on the NTSC system and were standardized in the 1960s. Because of the different electric current used overseas, PAL and SECAM use only 50 fields per second, but 625 lines per field. Therefore, European and Asian video has more resolution on every video frame than American video.

High-definition televisions (HDTV) systems offer even more resolution. Each system has a different resolution and color palette. The NTSC system is the oldest and most restrictive, so images that are edited for NTSC usually look good on the other systems as well.

Designing for Printers

Although no one uses a printer to surf the Web, many users print Web pages that contain a lot of text. On a printed Web page, the quality of elements such as navigation bars is not important. However, figures that accompany the text do need to be clear, as do logos and any advertisements. The main difference between an image displayed on a monitor and one that has been printed is the color system used. While monitors use RGB color, printers break primary colors into CMYK color space and use separate inks for cyan, magenta, yellow, and black. Because printers use different primary colors, and because printer inks have different limitations than the phosphors in a monitor, Web graphics often look different after being printed.

To preview what an image will look like when printed:

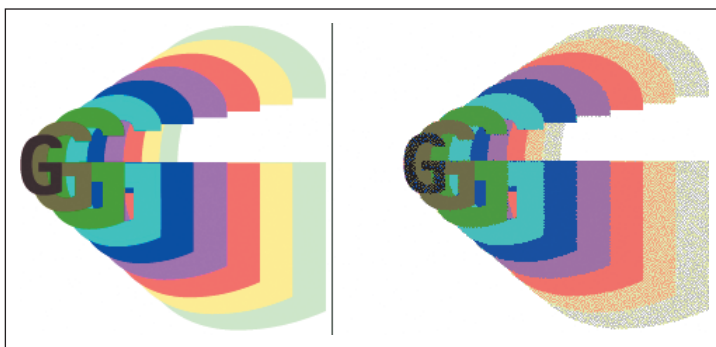
1. In Photoshop, open the **image** you want to preview.
2. Click **View** on the menu bar.
3. Point to **Proof Setup**, and then click **CMYK**.

The image appears as it will when printed on a color printer. Most colors appear darker, especially blues and greens. Reds appear more orange, and magentas and cyans can appear washed out.

The CMYK preview simulates printer output, but does not exactly match what you see when you print the image. Monitors can display only 24-bit color, which means about 16.7 million available colors. Printers use four primaries and are capable of 32-bit color, which is equal to 4,294,967,296 (over four billion) possible colors.

Printers also can dither images regardless of how you optimized them. What was an area of pale color might show up on paper as a few stray pixels. To solve this problem, avoid using very pale colors on important images such as site logos.

Figure 3-3 shows a Web graphic as it appears on screen and on paper with different, dithered colors.



An image as seen on a computer screen

The same image, when printed, shows dithering and uses different colors

Figure 3-3 A Web graphic as it appears on screen and on paper

Considering Operating Systems

Most Web surfers use some version of Windows, such as Windows 95, 98, or 2000. According to WebSideStory's *Statmarket.com*, in June 2000, 93.6% of Internet users

worldwide used a version of Microsoft Windows. Less than 5% use Macintosh, and an even smaller number use another type of operating system such as Linux or BeOS. Based on these findings, you should always design with a Windows audience in mind. This can be difficult, because if you're like most designers and graphic artists, you probably use Macintosh computers. Regardless of which system you use, be sure to preview your work in multiple environments.

The main difference between how Windows and Macintosh computers display graphics is that colors appear darker on Windows. The Windows operating system uses a lower **gamma** than Macintosh to display colors. Gamma refers to the point of middle-gray in a color system. This makes primary colors (reds, greens, and blues) appear richer on Windows than on Macintosh, and makes secondary colors (cyans, magentas, and yellows) appear richer on Macintosh than on Windows.

To see the difference in how Macintosh and Windows computers display colors:

1. In Photoshop, open an **image** file.
2. Click **View** on the menu bar.
3. Point to **Proof Setup**, and then click either **Macintosh RGB** or **Windows RGB** to see how the image appears on the different systems.

The other key difference in systems is how Macintosh and Windows PCs display text size. Macs display text about a third smaller than PCs. Mac text that appears about .5" tall appears about .75" tall on Windows.

As shown in Figure 3-4, text on a Windows computer (top) appears large and the primary colors are emphasized. On a Mac (bottom), primaries and secondaries are given equal weight and text is small.

As a designer, you might spend weeks preparing pages on your Mac, only to be shocked when you see how different they look on a Windows PC. The text differences are more of a concern to developers and producers than to graphic designers, though graphic labels and other text elements also are affected.

In general, design for Windows users, because they account for about 90% of your users.

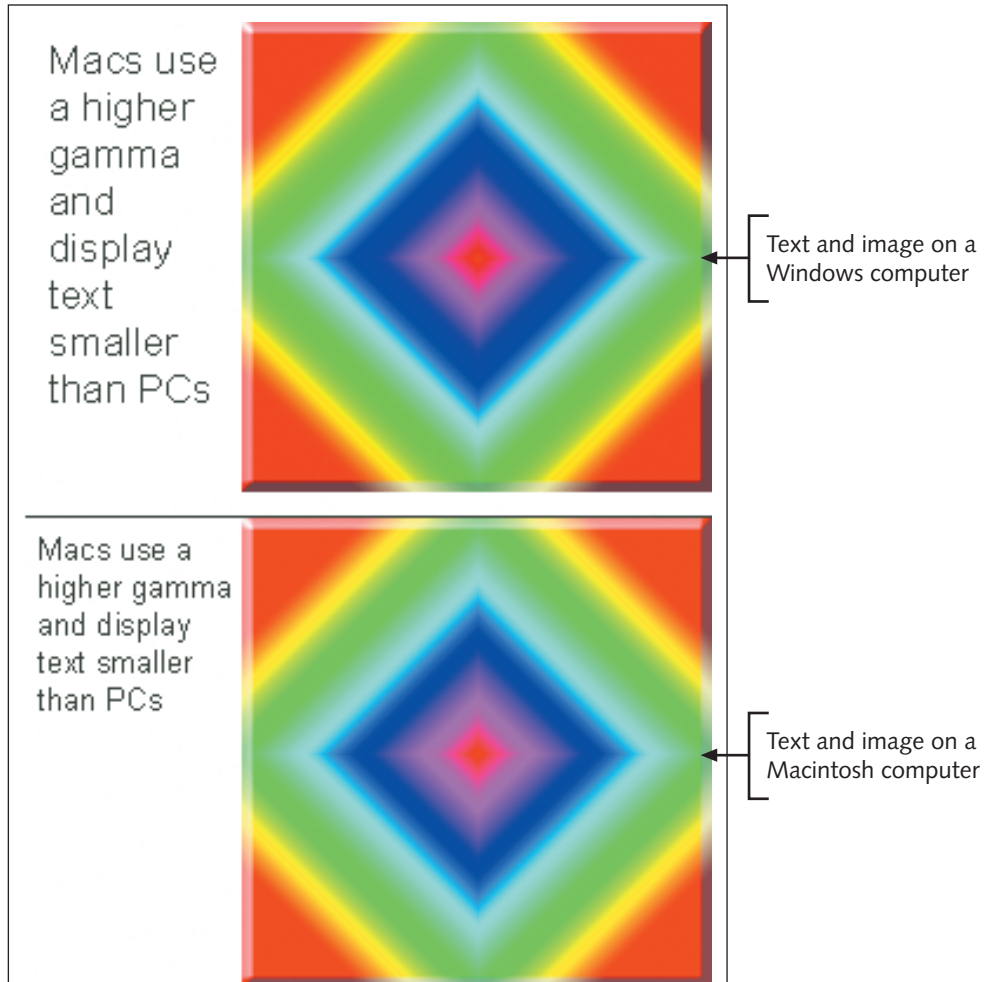


Figure 3-4 Text and colors as they appear on Windows and Macintosh systems

Considering Browsers

Just as the Microsoft operating system dominates the platform market, its Web client, Internet Explorer, also dominates the browser market. According to WebSideStory's *Statmarket.com*, in June 2000, 86.1% of Internet users worldwide used a version of Microsoft Internet Explorer. Its closest competitor is Netscape Navigator, which once was used by a majority of Web users, but now is used by less than 20%. Other brands of Web browsers represent an insignificant percentage of users.

The two main browsers display Web pages differently, but these differences all involve interpreting and rendering HyperText Markup Language (HTML). There is no significant difference in how they display graphics. However, the one difference you need to be aware of is how they display HTML colors. Almost any color you define in HTML might appear slightly darker in one browser than in another. In fact, of all 16+ million colors possible in the 24-bit RGB color space, only 216 are guaranteed to appear the same in both browsers. These colors make up the Web-safe color palette and are the subject of the next section. In truth, these 216 colors are not absolutely certain to appear the same in both Netscape and Explorer. Even different versions of Explorer can display these colors with slight differences. In general, however, if you use colors from the Web-safe color palette, your images will look the same regardless of the browser used.

DEFINING COLOR ON THE WEB

Web graphics are simply grids of colored pixels. They have no actual edges or textures, and the appearance of images is controlled entirely by varying the colors in the pixels. To create and use Web graphics, then, requires a thorough understanding of how people see color as well as how it is created and used on the Web.

Every color is a mixture of different amounts of red, green, and blue. For Web graphics, colors are defined in a special syntax called hexadecimal notation. In this section you will learn basic color theory and how to define colors on the Web.

Understanding Color Theory

You probably have worked with primary colors before. With paint, the primary colors are red, yellow, and blue. From these three paint primaries, you can create nearly any color. Initially it's easiest to create the three secondary colors: red and yellow make orange, yellow and blue make green, and red and blue make purple. You can then make more subtle shades by adjusting the amounts of the component colors. This is how colors are created on computer screens as well, although the primaries used are not the same as they are with paint.

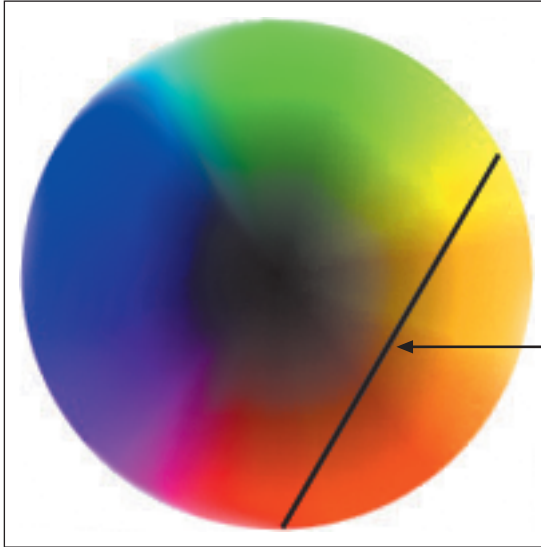
There are a few differences between painting on canvas and defining colors on a computer screen. These differences are discussed in the following sections.

Using Additive Versus Subtractive Colors

With paint, you apply a color directly to a white canvas or piece of paper. Each color you add darkens the color value. The orange you create by mixing red and yellow is darker than either component color, as shown in Figure 3-5. If you mix all the colors, you create a color that is nearly black. This method of color mixing is known as **subtractive color** because each additional color lessens the lightness of the final color.



When mixing all colors, you don't create true black because the pigments in paint are not true colors. Instead, paint pigments rely on chemicals and metals that only approximate what our eyes see as pure colors.



When using inks or paint, the orange color, halfway between pure red and pure yellow, is darker than either of the components colors

Figure 3-5 Paint uses subtractive color

With paint, you create new colors by mixing primary colors together. As more colors are added, the resulting color becomes darker. Computers also create new colors by mixing primary colors. However, as more primary colors are added in a computer graphic, the resulting color becomes lighter. Unlike paints, which generate colors through pigments, computer screens create color with light. A computer monitor shines a color on the screen, and each addition of color brightens the value of the overall color. This is known as **additive color**. Adding no color leaves a dark screen, and adding all colors makes the screen white. On a computer screen, the secondary colors are twice as bright as the primary colors, while with paints the secondary colors are half as bright, as shown in Figure 3-6.

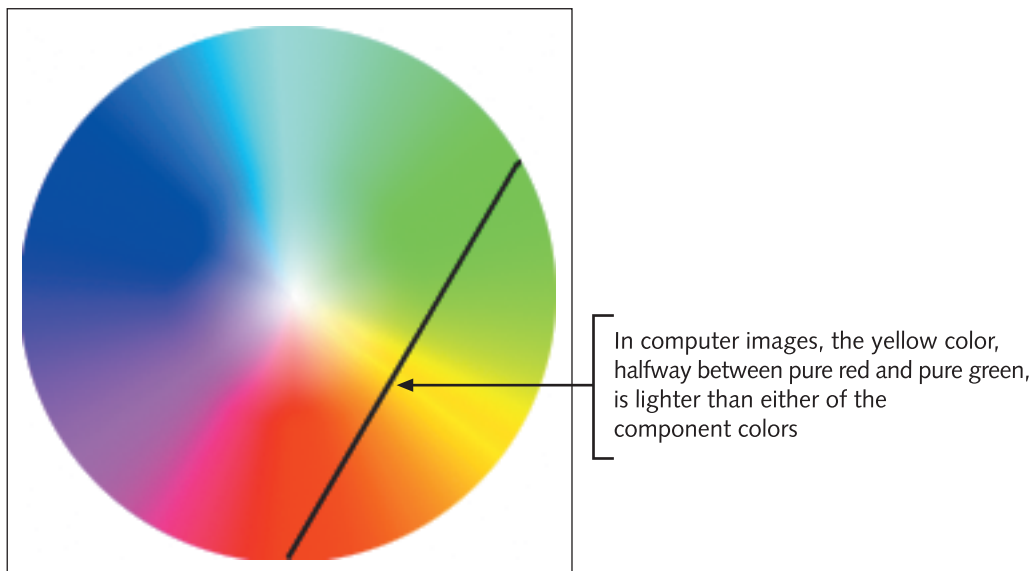


Figure 3-6 Computer images use additive color

Using Different Primary Colors

The methods of mixing colors for paint and pixels are different, and so are the primaries they use. For paint, the primaries are red, yellow, and blue, while for pixels they are red, green, and blue. Theoretically, almost any three colors can be used as primaries, as long as they are spaced evenly around the color wheel. As mentioned earlier, most printers use the CMYK color system, which has cyan, magenta, yellow, and black as primaries. However, cyan, magenta, and yellow are usually regarded as secondary colors in computer graphics.

Just as red and blue make magenta on your monitor, cyan and magenta also can make blue on a printer. In both cases, mixing two colors generates a third color which appears between them on the color wheel. Green can be used as a primary in paints, but if you mix it with red, trying to make yellow, you create a very dark shade of yellow. The yellow is only half as bright as the green and the red used to make it. Paints rely on red, yellow, and blue because of the subtractive nature of mixing colors with paint. Because the color darkens with every addition of color, you need to start with the brightest colors you can. Yellow is the brightest color our eyes can see so it makes sense to use that as a primary when using paint. However, yellow is not inherently brighter than other colors. It is only because of the sensitivities of our eyes and brains that yellow appears so bright. The way people see colors affects how color is used in graphics.

Understanding the Human Eye

A computer screen's primary colors more closely approximate the natural primaries that our eyes use. In the backs of our eyes, the retina senses light. The retina is a membrane composed of several layers, including one layer that contains rods and cones. The long, thin rods are sensitive to white light. The cones are sensitive to three colors: red, green, and blue. Our eyes work like a computer monitor: when there is no light, we see black. When we see combinations of colored light, different cones are stimulated and we see the combined color. For example, when we see yellow light (either emitted from a light source, or reflected from a yellow surface) the red-sensitive cones and the green-sensitive cones are stimulated together, telling our brain that we are seeing yellow.

Design often seems arbitrary. Beauty is said to be in the eye of the beholder. In fact, much of what looks good or bad is based on the physical structures of our eyes and brains.

Using Complementary Colors

Retina cones work in pairs. Some cones detect either red or green. This is why there is no reddish-green or greenish-red, and also why red and green are **complementary** colors. Complementary colors are opposite each other on the color wheel, stimulate completely different parts of the retina, and have high contrast when seen together. The complementary color for blue is yellow, which results when the retina's red- and green-sensitive areas are stimulated. Blue text against a yellow background is much easier to read than blue text against a purple background, for example.

Blue and purple are **analogous** colors. Analogous colors are near each other on the color wheel, stimulate similar areas of the retina, and have low contrast when seen together. Orange, red, and pink also are analogous colors. Recall that mixing two primaries creates secondary colors. Mixing primaries and secondaries creates **tertiary** colors. Tertiary colors are analogous to the adjacent primary and secondary colors. Figure 3-7 shows complementary and analogous colors.

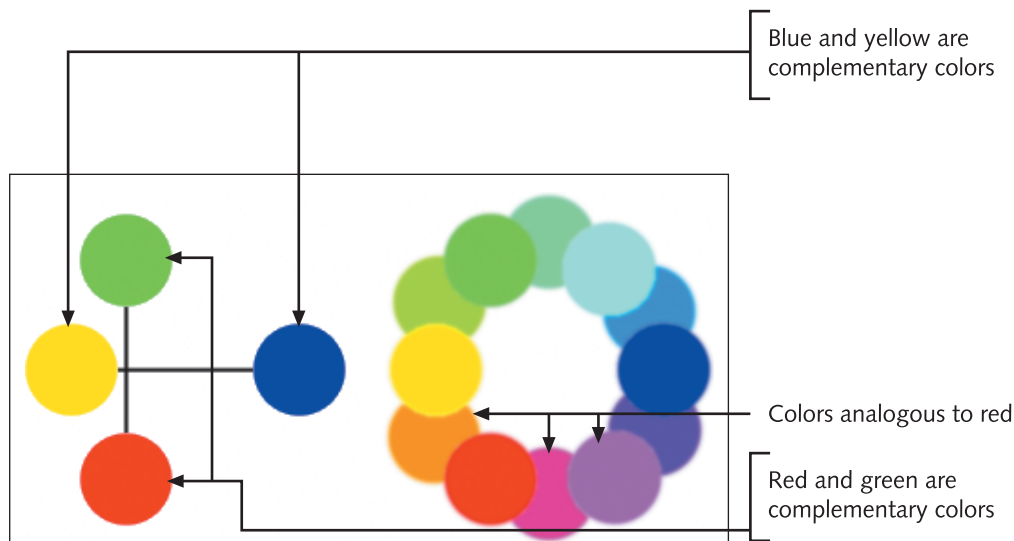


Figure 3-7 Complementary and analogous colors

The red and green retina cones are more sensitive than the blue cones, so blue is usually seen as a darker color. Even if the amount of blue light is the same as other colors, we see it darker because our eyes are less sensitive to blue than to red or green. This also explains why secondary colors appear brighter than primary colors in computer images. Yellow, for example, stimulates both red and green in the retina: In fact, it stimulates them twice as much as red or green alone. Because we are more sensitive to red and green than to blue, the mixture of red and green (yellow) appears brighter than the mixture of red and blue (magenta) or the mixture of green and blue (cyan).

When you design and position graphics in Web pages, be aware of whether the colors are analogous and have low contrast with each other, or whether they are complementary and appear bright. If you want colors to stand out, use them with a background that has complementary colors.

Understanding RGB Color

In a computer's RGB color system, each primary color has a channel, and each channel can take a value. Combining the channels determines the color. For example, 0% Red, 0% Green, and 0% Blue creates black, while 100% Red, 100% Green, and 100% Blue creates white.

Imagine that each of the three color channels could only be 0% or 100%; that is, only on or off. Then each color has two possible states, or only one bit of information is associated with each color. Because a computer has three color channels, with one bit per channel, it creates a 3-bit color system ($2^3 = 8$), providing eight possible colors.

Table 3-1 Colors created by combining the three primary colors

| Possible colors | Red channel | Green channel | Blue channel |
|-----------------|-------------|---------------|--------------|
| Black | 0% – off | 0% – off | 0% – off |
| Red | 100% – on | 0% – off | 0% – off |
| Yellow | 100% – on | 100% – on | 0% – off |
| Green | 0% – off | 100% – on | 0% – off |
| Cyan | 0% – off | 100% – on | 100% – on |
| Blue | 0% – off | 0% – off | 100% – on |
| Magenta | 100% – on | 0% – off | 100% – on |
| White | 100% – on | 100% – on | 100% – on |

If a computer allotted four bits of color information per pixel, you would have 2^4 , or 16 possible colors. In fact, the standard Windows palette is a 4-bit color system.

Eight bits per pixel provides 8-bit color, or 256 colors, and 24-bit color provides eight bits per color channel. Three channels provides 16,777,216 possible colors (2^8 Red, 2^8 Green, and 2^8 Blue gives 2^{8*3} , which equals 2^{24}).

Understanding the Color Cube

You can visualize any color system as a cube. Each of the eight corners is made up of a primary color, a secondary color, black, or white, as shown in Figure 3-8.

**Figure 3-8** An 8-color cube

Now suppose that instead of just on and off, you could indicate different levels of color for each of your three primaries. Instead of just 0% or 100%, you could allow each color to have gradations of value in 20% increments. If each color were 0%, 20%, 40%, 60%,

80%, or 100%, then you would have six possible states for each channel. Three channels each with six possible states can combine into 216 colors. Just as two states for three colors gives $2^3 = 8$ colors, six states for three colors gives $6^3 = 216$ colors. These colors are known as the Web-safe color palette.

In this system, you can combine different values of colors. For example, 100% (or full) red, 0% green, and 0% blue yields pure red, and 100% red, 100% green, and 0% blue yields pure yellow. To create orange, which is between red and yellow, you need to lower the amount of green. For example, 100% red, 60% green, and 0% blue yields a yellowish orange; reducing green further allows red to dominate the color.

Figure 3-9 shows how full red with no green or blue results in pure red. Full red with full green and no blue results in pure yellow, and full red with 60% green and no blue results in a yellowish orange.

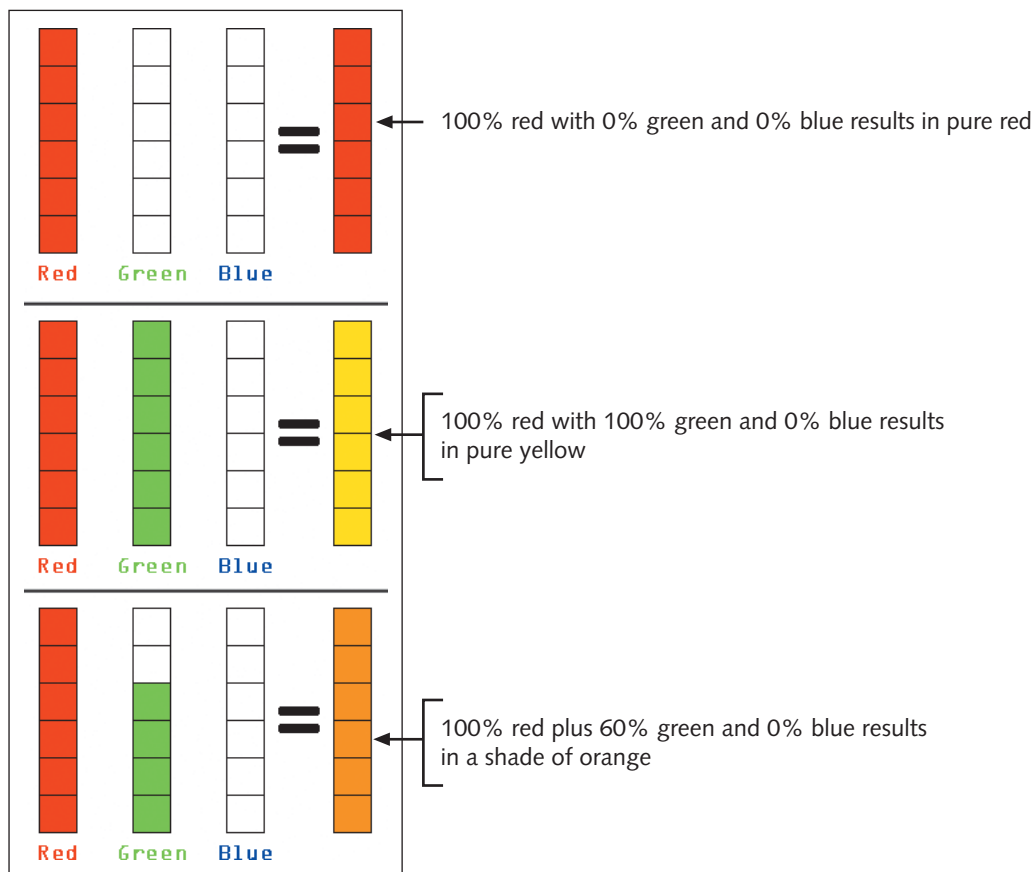


Figure 3-9 Mixing primaries to create different colors

Using Hexadecimal Notation

Often you do not know the percentage of color per channel, but you do know the value compared to 255. There are 256 colors per channel, but for HTML colors, the counting starts at zero, so the entire scale is from 0–255. A value of 0 for each channel yields black, while a value of 255 for each channel yields white.

Some software describes the relative amounts of the three primaries as percentages, as described in the previous Using the Color Cube section. On the Web, however, you need to use hexadecimal notation because a percent system cannot represent all the colors when each channel has 256 possible values.

To specify colors in HTML, you use a special syntax known as hexadecimal notation. Colors defined this way always have six characters and look like random series of numbers and letters, such as #0c48fb or #e62a15. However, hexadecimal notation is a system of highly ordered regular numbers that are easy to understand and use once you are familiar with their structure.

To understand a hexadecimal color, first break it down into three pairs. Each pair of numbers defines one RGB color channel. For example, in the number #048cfb, “04” defines the red value of the color and is equivalent to a value of 4 out of 256 when looking at the Color Picker dialog box in Photoshop. The hexadecimal value “8c” defines the green value and is equal to a value of 140 out of the 256 colors in the green color channel. The value “fb” defines the blue value and is equal to 251.

Counting systems each have a base number at which the counting repeats. The simplest base is base 1, where you use tick marks to indicate numbers. Base 2 is used in computer memory, where each unit has one of two values, 0 or 1. Hexadecimal means that the counting system is base 16.

Bases less than 10 simply eliminate certain digits from counting. In octal notation, which uses eight as the base, once the counting passes seven, you start counting again. Bases larger than 10, however, present counting problems, because there are not enough digits to express numbers such as 11, 12, and so on, in these bases. Bases larger than 10 use letters to express values greater than 10. Because the digits 1 and 0 (10) equal the value 16 in base 16, you need another character for the value of 10; hexadecimal notation uses the character A. For 11, it uses B, for 12 it uses C, and so on until 16. At that point the digits move to 16 and the counting starts again at 10, which is equivalent to 16 in decimal notation.

In base 10, the ones column repeats every 10 numbers. In base 16, the ones column repeats every 16 numbers. Table 3-2 shows equivalent values for three different counting systems: octal, decimal, and hexadecimal.

Table 3-2 Equivalent values for different counting systems

| Base 8 — octal | Base 10 — decimal | Base 16 — hexadecimal |
|----------------|-------------------|-----------------------|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| 4 | 4 | 4 |
| 5 | 5 | 5 |
| 6 | 6 | 6 |
| 7 | 7 | 7 |
| 10 | 8 | 8 |
| 11 | 9 | 9 |
| 12 | 10 | A |
| 13 | 11 | B |
| 14 | 12 | C |
| 15 | 13 | D |
| 16 | 14 | E |
| 17 | 15 | F |
| 20 | 16 | 10 |
| 21 | 17 | 11 |
| 22 | 18 | 12 |
| 23 | 19 | 13 |
| 24 | 20 | 14 |
| 25 | 21 | 15 |
| 26 | 22 | 16 |
| 27 | 23 | 17 |
| 30 | 24 | 18 |
| 31 | 25 | 19 |
| 32 | 26 | 1A |
| 33 | 27 | 1B |
| 34 | 28 | 1C |
| 35 | 29 | 1D |
| 36 | 30 | 1E |
| 37 | 31 | 1F |
| 40 | 32 | 20 |
| 41 | 33 | 21 |
| 42 | 34 | 22 |

To use hexadecimal notation to describe the 255 Web colors, you need to know the relative values of red, green, and blue, and convert the values to hexadecimal.

Figure 3-10 shows a color that is 20% red, 80% green, and 60% blue. In hexadecimal, this is expressed as #33CC99. (Interpreting this number is explained below.) The number #33CC99 represents a Web-safe color because all values are in 20% increments, such as 0%, 20%, 40%, 60%, 80%, or 100%. In hexadecimal notation, these values translate to: 00, 33, 66, 99, CC, and FF, respectively. The case does not matter; cc is the same as CC.

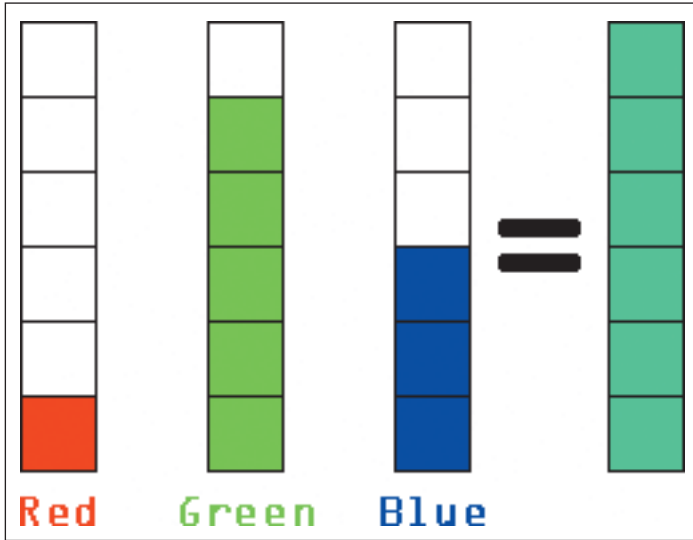


Figure 3-10 The color expressed by #33CC99

A good way to practice adding colors together is with the Color Picker dialog box in Photoshop. You can combine different values for red, green, and blue to generate different hexadecimal colors.

Each primary color has six Web-safe values that can be represented as percentages, fractions of 255, or as hexadecimal numbers. Table 3-3 shows the equivalent color values for the three common units: percentages, decimals between 0 and 255, and hexadecimals between 0 and FF.

Table 3-3 Equivalent Web-safe color values for different units

| Percentages | Out of 255 | Hexadecimal |
|-------------|------------|-------------|
| 0% | 0 | 00 |
| 20% | 51 | 33 |
| 40% | 102 | 66 |
| 60% | 153 | 99 |
| 80% | 204 | CC |
| 100% | 255 | FF |

To interpret a hexadecimal color value such as #1f6d9c, start with the first value. In this example, 1f is the red value, which is 31 in decimal, and under 20% of 255, which is 51. That means the red value for this color is low.

The middle two digits—6d—represent the green value. This value is higher than 66 (hex) or 102 (decimal), which is 40% of 255. The color therefore is 40% green. The blue value, 9c, is higher than 99 (hex) or 153 (decimal), which is 60% of 255. You can then determine that the color has almost no red, 40% green, and 60% blue—a gray-greenish blue.



The pound sign (#) often precedes hexadecimal numbers. The pound sign is an optional notation indicating a numerical value.

Interpret another hexadecimal value: #F93C60. The easiest way to decipher the color is to round each number pair to its nearest hexadecimal number. Break it down into pairs: F9 – 3C – 60, and round to the nearest Web-safe values: FF – 33 – 66. The values are the same as 100% red, 20% green, and 40% blue, or red with a shot of hot pink. These color additions are shown in Figure 3-11. Remember that color mixing on the Web is additive, so by taking full red and adding a little green and blue, the result is a lighter color than the full red alone.

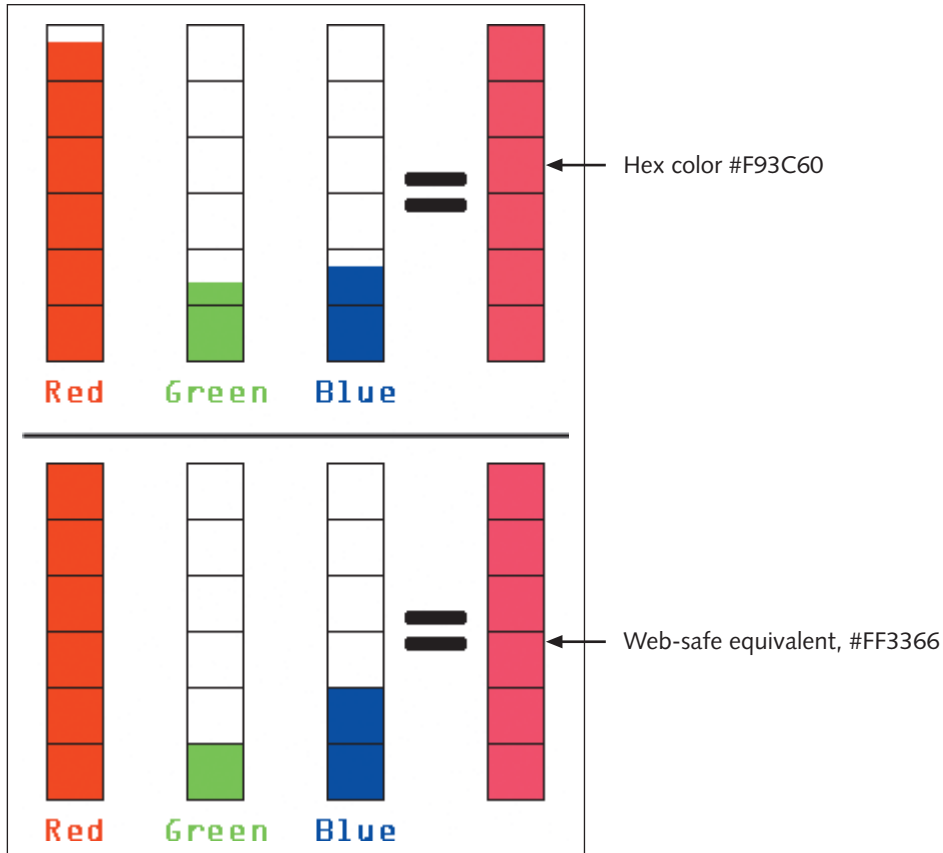


Figure 3-11 Hexadecimal color #F93C60 and its nearest Web-safe equivalent, #FF3366



The complete chart of all Web-safe colors and their hexadecimal codes is in the Appendix.

Using Color Names

Many people have trouble thinking of colors in terms of numbers, and prefer to use color names as much as possible. You can use names to identify colors in a browser, but not all will work. Netscape understands more names than Internet Explorer, including names such as papayawhip and darkgoldenrod (both are shades of yellow). However, Internet Explorer interprets the individual letters in names as hexadecimal characters, and produces colors other than what you intend. If you do use names, use the names of colors that both browsers understand. Only the 16 most common color names are understood by all browsers. These color names and their associated hexadecimal values are shown in Tables 3-4 and 3-5.

Table 3-4 The eight Web-safe color names

| Color Name | Hexadecimal Value |
|------------------|-------------------|
| Black | 000000 |
| Red | FF0000 |
| Yellow | FFFF00 |
| Lime | 00FF00 |
| Cyan, aqua | 00FFFF |
| Blue | 0000FF |
| Magenta, fuchsia | FF00FF |
| White | FFFFFF |

Table 3-5 The eight non-Web-safe color names

| | |
|--------|--------|
| Gray | 808080 |
| Maroon | 800000 |
| Olive | 808000 |
| Green | 008000 |
| Teal | 008080 |
| Navy | 000080 |
| Purple | 800080 |
| Silver | C0C0C0 |

Using Web-Safe Colors

While #1F6D9C and #F93C60 are not Web-safe colors, they still appear properly in most environments. With blocks of solid color, such as backgrounds, the browser displays the closest approximation for the color. This approximation is different in different browsers, however. If you use a color with an intermediate value such as 80, one browser might round down to 66, and another might round up to 99. With images, the browser might also dither the non-Web-safe colors to approximate the colors.

Photoshop allows you to save images with the Web-safe color palette. This shifts all colors in the existing palette to the nearest Web-safe equivalent. However, you do not have to save images with the Web palette to have them appear in a browser. In fact, if you do, the image will probably become heavily banded. As with any color-reduction method, high-color images with color gradients tend to look distorted when reduced to the Web-safe palette.

The best time to use Web-safe colors is when you want to display an image on a colored background. The background color of the page and the background color of the image must both be the same Web-safe color, or the boundary between image background and

page background might be visible. When you use Web-safe colors, the colored background of an image matches the background color of the page. Figure 3-12 shows an image displayed on a colored background. Both background colors are set to the same non-Web-safe color. The browser displays the two colors differently, and the seam between them becomes visible.



Figure 3-12 An image on a colored background using non-Web-safe colors

CONTROLLING WEB GRAPHICS WITH HTML

As a graphic artist, you can accomplish most of your HTML coding with a **what you see is what you get (WYSIWYG)** editor. To really understand page layout, however, and appreciate what the editor does for you, you need to understand the HTML code itself. You can use a number of markup languages to create Web pages. The most common is HTML, but you also need to be aware of CSS (Cascading Style Sheets), CSS-P (Cascading Style Sheets-Positioning), and XML (Extensible Markup Language).

As a graphics professional, you probably only need to know HTML. An HTML tag can have up to four types of elements. The basic structure of a tag is:

`<TAG ATTRIBUTE="value"></TAG>`

- The **tag** itself is surrounded by angle brackets and tells the browser to place something on the page. The image tag is written as ``.
- Most text formatting tags require closing tags to indicate what text is being affected. The IMG tag, however, does not require a closing tag.
- The **attributes** are placed within the angle brackets after the tag name. You can have many attributes in one tag, and the order of attributes does not matter. Some of the attributes of the IMG tag are SRC, WIDTH, and BORDER.

- Most attributes have **values**, surrounded by quotation marks. For example, in the HTML code ``, SRC is the attribute and myImage.jpg is the value. Browsers usually treat the quotation marks as optional, meaning they understand the HTML whether or not you include them. However, future browsers are expected to require them, so it is good practice to always include quotation marks around values.
- The case of HTML tags does not matter. The tags and attributes can be written in uppercase or lowercase. (In these examples, tags and attributes are written in uppercase to aid readability). Filenames, however, are an exception. Some servers require the image filename and the value of the SRC attribute to match exactly, or they will not display the image.

Choosing HTML Editors

You have a few options for generating the HTML you need to make pages and display your images on the Web. Most Web production teams include a producer who is responsible for page layout, but graphic designers also often create Web pages. A popular tool you will likely use to create Web pages is a WYSIWYG editor such as Dreamweaver.

Using WYSIWYG HTML Editors

WYSIWYG refers to software with an interface that resembles the final output. A word-processing program is essentially a WYSIWYG text editor, because what you see on the screen is similar to what you get when the document is printed. Bold text appears in boldface on the screen, and different fonts and sizes appear on screen as they appear on paper. Plain HTML typically works differently. Text that appears in bold in a browser appears as plain text surrounded by bold tags in an HTML document. You can insert these tags by typing them in a text editor, or you can use software that lets you lay out pages without having to learn HTML. Although it is sometimes easier to learn to use a WYSIWYG editor than to learn all the HTML tags, the editors do not allow the same control as hand-coding HTML.

Most Web graphic artists and designers use the Macintosh as their OS, Netscape as their browser, and Dreamweaver as their HTML editor. The disadvantage to this setup is that these designers rarely see their work in the same way that most of their audience does. The majority of their audience uses a different operating system and browser. The advantage to this conflict is that it usually results in better cross-platform products. Web graphic artists and designers could primarily use Microsoft Windows as the OS, Internet Explorer as the browser, and FrontPage as the HTML editor. Because Microsoft tools work together in ways that tools from disparate companies cannot, Web pages developed with this approach are not as compatible in non-Microsoft environments. The major criticism of FrontPage is that it sometimes creates pages that work when viewed on Windows with Internet Explorer but don't lay out properly on other systems.

Using Straight Text Editors

Web purists claim that only hand-edited HTML code is good enough for creating Web pages, and for complex layouts they are correct. Even the best WYSIWYG editors cannot create code as clean or layout as pixel-perfect as you can with a text editor. A text editor is a program that edits only **ASCII text**. ASCII text is the list of 256 basic characters used in computer files. ASCII text includes all uppercase and lowercase letters, digits, punctuation marks, and various European letters. It does not include information about font or size.

You can use text editors to write programs in C or Java, and to write code in JavaScript, Cascading Style Sheets, XML, HTML, or just about any other computer language. Unlike word processors, which add hidden characters that format the text, text editors require you to explicitly state formatting code, usually in the form of tags. Most developers who code HTML by hand use Windows Notepad. The most popular commercial text editor for Mac users is BBEdit.

Using an HTML Converter Application

An HTML converter is a program that allows you to format text and add graphics in one application, and then converts the entire document to a Web page. Microsoft Word is the most common HTML converter; it allows you to save any Word document as HTML.

Converters often add unnecessary tags and characters to Web pages, resulting in pages that take longer to load than necessary. Converted pages also usually have poor layouts, with gaps between images and misplaced elements. Until converter applications overcome these weaknesses, use HTML converters only as a last resort, when no other tool is available.

Whether you use an HTML editor or code by hand, you need to be familiar with the HTML tags, including IMG and TABLE, used to display images in Web pages. These tags and their common attributes are discussed in the following sections.

Using the IMG Tag

The HTML tag used most often to add graphics to Web pages is the image tag: . Though most HTML tags require a closing tag, does not. This tag controls how images appear by using a variety of attributes, which are discussed in the following sections. Images can stand alone or be used as links. To use an image as a link, place an anchor tag, <A> and , around the image tag, as in the following code:

```
<A HREF="page2.html"><IMG SRC="myImage.jpg"></A>
```

This displays the image myImage.jpg and links it to page2.html.

Once a browser loads an image, it does not need to load the image again. Every file requested by the browser (whether HTML code, images, java, and so on) is stored in the browser's **cache** directory. The cache is a folder of previously viewed files that can be reused without retrieving them again from the server.

The advantage to caching is evident when you use the same image more than once. The first time the image is used, it is downloaded from the server, but each successive time it is accessed from the browser's cache, significantly reducing the time it takes to display the image.

For example, if you have a logo in the upper-left of every page, the logo image is loaded from the server only the first time the user visits a page. As the user visits additional pages, only the new images are loaded from the server.

The effect produced by the IMG tag is controlled by the following attributes: SRC, LOWSRC, WIDTH, HEIGHT, HSPACE, VSPACE, ALIGN, ALT, and BORDER. These attributes are described in the following sections.

Defining an Image with the SRC Attribute

The source attribute SRC tells the browser where to retrieve the image file. This is the only attribute required by the tag. The value of the SRC tag includes the file path and the filename. The file path is the list of folders leading to the filename. A file path can be **relative** to the HTML file that contains the reference, or it can be **absolute**, meaning it is relative only to the root directory of the Web server. For example, tells the browser to find a folder called images in the same folder as the HTML file. The browser finds a folder called new in the images folder, and then finds the file named myImage.jpg. The file path images/new/myImage.jpg is relative to the HTML file that contains the tag.



The word directory is interchangeable with the word folder.

An example of an absolute file path is: . In this case, the forward slash (/) at the beginning of the path indicates that the path is absolute, or relative only to the root directory. The reference is the same as in the other example, except the path begins at the root rather than at the HTML file's location.

For a path that travels up the directory tree, rather than down, use the following notation: ../(period, period, slash). This tells the browser to look in the parent directory of the current location. For example, tells the browser to look up one directory, then back down to httpd, then to the images directory, to the new directory, and then find the file.

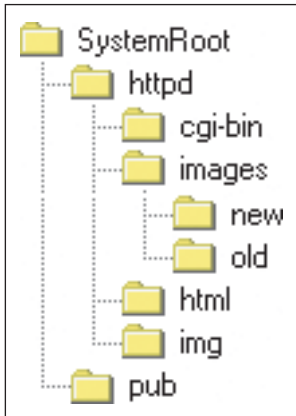


Figure 3-13 A sample directory structure

You also can indicate an absolute file path. This means the path is relative to the server root. The server root is the point from which all Web file paths originate. In the structure in Figure 3-13, the server root is the directory called `httpd`. Anything in the `pub` directory will not show up on a Web browser. The server root is different from the system root, which is the bottom-level directory on the server that contains all other directories.

Defining an Alternate Image with the LOWSRC Attribute

A Web page with large images might appear slowly as the browser waits for the images to load from the server. To solve this problem, you can use the low-source attribute `LOWSRC`. You can use `LOWSRC` to load a smaller version of an image first; that image is then replaced with the normal version. If you have an image of 120K, for example, you might not want the browser to wait. You can create a low-quality 2K version and set the `LOWSRC` attribute to show that version, as in the following code.

```
<IMG SRC="high-color.jpg" LOWSRC="low-color.gif">
```

The low-quality image loads first, then the rest of the page, and finally the high-quality version. You can use any image for the `LOWSRC` image.

Controlling Image Dimensions with the WIDTH and HEIGHT Attributes

You can specify the size of the image display with the `WIDTH` and `HEIGHT` attributes. If you want a 10 × 10 pixel image to appear as 36 × 72, you would use the following code:

```
<IMG SRC="myImage.jpg" WIDTH="36" HEIGHT="72">
```

However, browsers do not interpolate pixels the way image editors do. Using the code above, you end up with a blocky, pixelated version of the original. Figure 3-14 shows how using HTML to stretch an image results in jagged edges.



Figure 3-14 Stretching an image with HTML

You also can use percentages as widths. Setting the width to 100% stretches the image to the right of the browser window.

In general, you should always include the height and width attributes and set them to the actual size of the image. The browser displays the page more quickly if it already knows the dimensions of the images.

Creating Thumbnail Images Using WIDTH and HEIGHT Attributes You also can force the image to appear smaller than its actual size by using smaller values for the height and width. If the original image is 640×480 , for example, you can show a **thumbnail** version of the image. Thumbnails are small versions of a larger image, and usually are used as links to the full-size image. Thumbnails are discussed in detail in the Creating Thumbnail Galleries chapter.

One way to use thumbnails is to create small duplicates of every file, and then use those duplicates as links to the larger images. The other way is to use the large images themselves as the links, but scale them down with the height and width attributes.

To use duplicated thumbnails as links, you must create additional graphics for each thumbnail. When you use scaled thumbnails as links, all of the full-size images have to load completely even though you are using them only as buttons. However, once the images are loaded, they do not have to load again and they appear immediately when requested.

To illustrate the difference, imagine you have a gallery of 20 images, each one around 125K. If you make a 5K thumbnail version of each image, your gallery page has 100K (20 images at 5K) worth of graphics and loads in a few seconds on most modems. As each full-size image is requested, it is loaded from the server.

If your thumbnails are not small images, but are scaled-down versions of the actual images, the thumbnail page has 2,500K ($125K \times 20$ images) worth of graphics to download. This takes a significant amount of time, even on fast connections. However, once all the thumbnails are loaded, the full-size images do not need to be loaded again because they are really the same images.

In general, small duplicates of images used as thumbnails provide quicker user access than pages with scaled down images.

Adjusting Image Margins with the HSPACE and VSPACE Attributes

You control the margins around images with the HSPACE (horizontal space) and VSPACE (vertical space) attributes. The values for these attributes are numbers which tell the browser how many pixels to pad around the image. Following is a line of HTML code using the HSPACE and VSPACE attributes:

```
<IMG SRC="myImage.jpg" HSPACE="5" VSPACE="0">
```

In this example, the browser pads 5 pixels on both sides of the image, and does not pad the top or bottom.

Positioning Images with the ALIGN Attribute

You use the ALIGN attribute to control where the image is positioned relative to the other elements. By default, an image appears with its bottom aligned to the baseline of adjacent text. You also can set this attribute to several different values:

- *Top*: Sets the top of the image to the top of the adjacent text
- *Bottom*: Sets the bottom of the image to the baseline of the adjacent text
- *Center or Middle*: Centers the image (Netscape and IE display centered images differently.)
- *Right and Left*: Positions the image on the right or left side of the browser window and allows adjacent text to wrap around the image

In Figure 3-15 the top image is not aligned and has a margin of 0 pixels, set by the HSPACE and VSPACE attributes. The middle image is left-aligned and has a margin of 5 pixels. The bottom image is right-aligned and has a margin of 10 pixels.

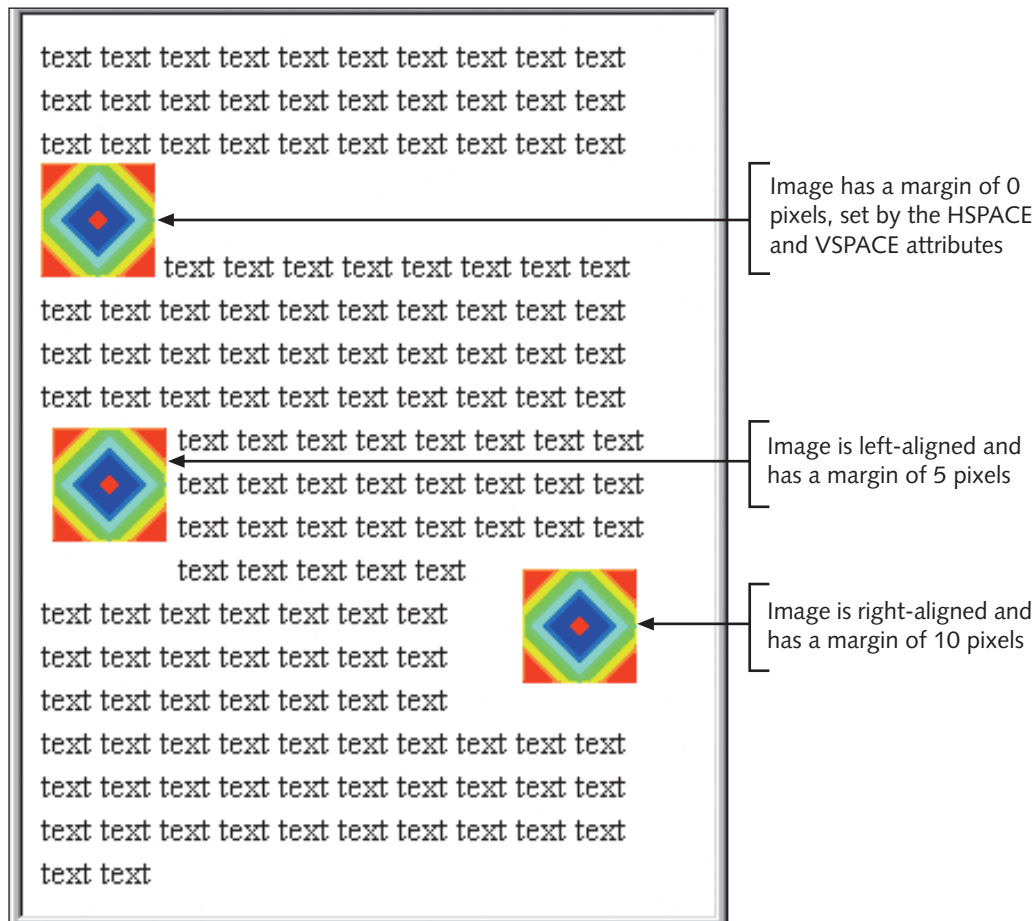


Figure 3-15 Examples of images with different attribute settings

Defining Alternate Text with the ALT Attribute

The value of the ALT attribute is that text appears when the image does not. Even after the image loads, the ALT text appears when the user points to the image. It is good practice to always include alternate text attributes in image tags.

ALT attributes can contain spaces and punctuation marks, as in the following example:

```
<IMG SRC="myImage.jpg" ALT="Drawing number 1">
```

The users who benefit from ALT attributes are:

- People who use text-only browsers such as Lynx
- People who have turned off image loading in their browsers to speed page loads

- People with slow connections and browsers that time out before all images are loaded
- Visually impaired users who use text-to-speech software to read Web pages

Another benefit of ALT attributes is that some search engines read them when deciding whether and how to index the page. Using the ALT attribute improves your site's ranking.

Setting the Image Border with the BORDER Attribute

Use the BORDER attribute to set a border for images you use as links. The default surrounds the image with a colored border equal to the LINK attribute set in the BODY tag (usually blue). You can turn off the default border by setting the BORDER attribute to 0, as in the following example:

```
<IMG SRC="myImage.jpg" BORDER="0">
```

Using the TABLE Tag for Layout

The table tag was originally intended only for the display of tabular data. Having no other method, developers began using tables to lay out Web pages as well. The version of Cascading Style Sheets called Cascading Style Sheets-Positioning (CSS-P, also referred to as layers) was the intended structure for layout, but even the newest browsers still do not support layers and positioning.

The TABLE tag encloses rows, defined by TR tags. Table rows enclose table delimiters (called table cells), defined by TD tags. All three tags can use the BGCOLOR attribute, which takes a hexadecimal color to define the background color, as in the following example:

```
<TABLE> <TR> <TD BACKGROUND="FF0000">top</TD> </TR> <TR>  
<TD>bottom</TD> </TR> </TABLE>
```

This HTML code defines a table with two rows and one cell in each row. The top row has a red background.

The TABLE tag can take a number of other attributes. The following three attributes affect the spacing of elements:

- The value of the BORDER attribute sets the width of the bevel on the ridge that surrounds the table and separates the table cells. Setting the value to 0 removes the bevel.
- The value of the CELSPACING attribute sets the width of the border between cells.
- The value of the CELLPADDING attribute sets the width between the border and the contents in the table cells.

Figure 3-16 shows two tables with two rows of two cells each. The table on the left uses the default values for BORDER, CELSPACING, and CELLPADDING. This results in

gaps between the images. The table on the right has all three attributes set to 0. This positions the images next to each other without margins.

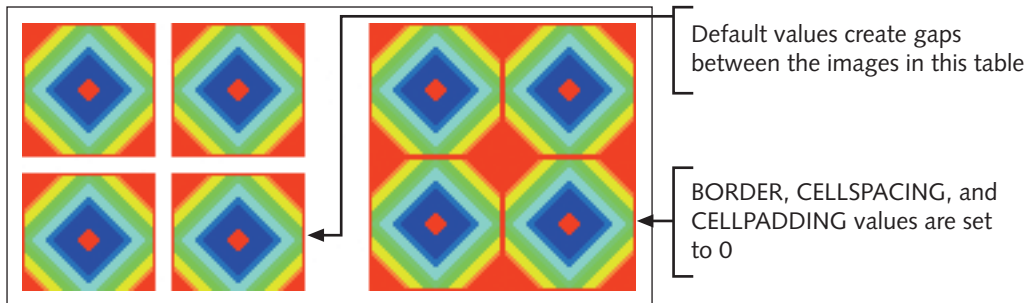


Figure 3-16 Images positioned in tables with and without margins

Using tables to lay out graphics is covered in the “Creating Splash Screens” chapter. For more information on table layout, see Joel Sklar’s *Principles of Web Design*.¹

AVOIDING COMMON MISTAKES

It’s easy to include images in a Web page, but sometimes the images don’t appear. The problem is that not all Web servers and browsers display images the same way; what looks good on your system might not load at all on another. The trick is to use HTML and filename conventions that are common to all systems and browsers. The following sections cover the solutions to the most common errors beginners make when using graphics in their Web pages.

Avoiding Unrecognized Characters in a Filename

When you save an image in an image editor such as Adobe Photoshop, avoid using unrecognized characters, including spaces, slashes, and apostrophes, so the Web server interprets the complete filename correctly. Most other punctuation marks also cause problems. In general, it’s safest to use only uppercase and lowercase letters, numbers, and underscores.

Avoiding Spaces

It’s often convenient to use a long, descriptive name such as `My favorite picture.jpg`, but the Web server stops reading the filename at the space. It then looks for a file called `My`. The solution for this problem is to use underscores (`_`) instead of spaces, as in `My_favorite_picture.jpg`.

¹ Sklar, Joel. 2000. *Principles of Web Design*. Boston, MA: Course Technology.

Avoiding Slashes

People often include dates in their filenames, such as WebCamPix-6/19/99.jpg. In this case, the Web server interprets the slashes as directory indicators. It looks for a file called 99.jpg in a directory called 19, which is itself inside a directory called WebCamPix-6. The solution here is to use a different syntax for dates. A better way to express dates on the Web is to indicate the year, month, and date in that order without punctuation. For example, the filename indicated above would be written as WebCamPix-19990619.jpg.

Avoiding Apostrophes

Another common mistake is to use apostrophes in filenames, as in My_dog's_portrait.jpg. In this case, the Web server interprets the apostrophe as a quotation mark and looks for a file called My_dog. Never use apostrophes or quotation marks in a filename.

Solving Format Problems

Web browsers can display images in only one of three computer graphic formats: GIF, JPEG, or PNG. If you have an image in the PICT format, for example, you cannot convert the image to the GIF format by simply changing the suffix to .gif. You must open the image in an image editor and save it in one of the three Web graphics formats. The basic steps for doing so in Photoshop are listed below. For more details see the “Optimizing Graphics for the Web” chapter.

To convert an image to a Web graphic format in Photoshop:

1. Open the **image** in Photoshop.
2. Click **File** on the menu bar, and then click **Save As**.
3. In the Save As dialog box, click the **Format** list arrow.
4. Click **CompuServe GIF (*.GIF)**, **JPEG (*.JPG, .JPE)**, or **PNG (*.PNG)**.

Solving File Suffix Errors

Web browsers rely on the image filename to define the image type. Specifically, they look at the suffix—the characters after the last period—to indicate the file format. If a filename doesn't include a suffix, the browser doesn't know the file type and ignores it. If a filename includes multiple periods, the browser looks only at the last suffix. For example, a file named my_image.gif.old appears to the browser as being of a format called OLD, and does not display it. If a file is named image.gif, the browser tries to display a gif image using the data in the file. If the file really is in the GIF format, it should appear. If it is actually in another format, such as JPEG, the browser tries to display the JPEG file as a GIF, and fails. Always include the proper format suffix when naming image files.

Avoiding Improper Capitalization

HTML tags do not need to be capitalized; `<IMG SRC=...` is the same as `<img src=...` which is the same as `<ImG sRc=...`. Web browsers ignore the capitalization of tags. However, it's common practice to capitalize all HTML tags to make them stand out from the rest of the text. Web server names can also be uppercase or lowercase. For example, `<A HREF="HTTP://WWW.GIF.COM/..."` is the same as `<A HREF="http://www.gif.com..."`.

On UNIX servers case does matter for filenames. If you save an image as `MyPicture.JPEG`, you must refer to the image using that capitalization in the HTML code. If you use ``, for example, the UNIX server cannot find the file. Even if you work only on Macintosh or Windows servers, do not get into the habit of ignoring case in filenames.

Avoiding Incorrect Path References

The errors caused by incorrect path references usually are the result of an imperfect understanding of the concept of file paths. To reference a file in the same directory, set the attribute's value to the filename, as in the following example:

```
<IMG SRC="myFile.jpg">
```

If the HTML file and the image file are in different directories, you need to indicate the path. For example, if you have an HTML file in the directory called `html` and an image file called `myImage.gif` in the directory called `images`, the HTML code looks like the following:

```
<IMG SRC="images/myImage.gif">
```

As you learned earlier, this is called a relative file path because the directory structure is relative to the location of the HTML file. The drawback to using relative file paths is that if you move the HTML file, you might have to change the HTML code.

The benefit of absolute file paths is that they are not related to the location of the HTML file. If you move the HTML file, you do not need to change the HTML code that calls the image.

Be careful not to use paths that are local to your hard drive; they will not work when you upload them to the server.

CHAPTER SUMMARY

3

- ❑ People use a variety of monitor sizes, and it is better that your graphics appear at a reduced size than to be cut off by the edge of the browser. Design graphics assuming they will be seen on a 15" desktop monitor.
- ❑ WebTV is gaining popularity, but represents a small minority of users. Design within the limits of television display only if you are concerned with how those users see your graphics.
- ❑ Most designers use Macintosh computers, but most Internet users use Windows computers. Make sure to preview your images on a Windows computer.
- ❑ Most users use Internet Explorer as their Web browser, but many also use Netscape Navigator. Preview your work on both browsers.
- ❑ Color appears differently on different monitors, operating systems, and browsers. To guarantee that colors in images match colors in HTML, use the Web-safe color palette of 216 colors.
- ❑ Web color is defined in RGB color, where each primary has a value represented in hexadecimal notation.
- ❑ Hexadecimal color is a way of describing 24-bit RGB color, where the value of each channel is defined by a two-digit number from 0 to F, and where the letters A through F represent the numbers 10 through 15.
- ❑ Hexadecimal colors have three pairs of numbers, one for each channel; each pair represents 256 different values.
- ❑ Color names are easier to use than hexadecimal colors, but only 16 color names show a similar color on both popular browsers, and only eight are Web-safe.
- ❑ You do not have to save images with the Web-safe palette unless you are concerned with how their backgrounds will blend in with the background of the page.
- ❑ WYSIWYG stands for “what you see is what you get,” and refers to software that displays your work the same way while editing as after output.
- ❑ WYSIWYG interfaces simplify the process of generating HTML but usually cannot lay out pages as accurately as hand-coded HTML.
- ❑ The basic structure of an HTML tag is `<TAG ATTRIBUTE="value"></TAG>`.
- ❑ You can use the IMG tag and its attributes to control how images appear in a Web page. The most common attributes for working with Web graphics are ALT, LOWSRC, ALIGN, HEIGHT, WIDTH, HSPACE, VSPACE, and BORDER.
- ❑ IMG tags should always include the HEIGHT, WIDTH, and ALT attributes.

- Using the TABLE tag is the primary way to control layout of Web pages. You use this tag to indicate background colors for tables, and especially to create effects without using images.
- Spaces, punctuation, and slashes in filenames cause problems with browsers and prevent images from appearing correctly.
- Image filenames must include the correct suffix.
- Windows and Macintosh computers ignore inconsistent capitalization of filenames, but most Web servers do not, and will prevent the display of images.
- Incorrect paths are one of the most common errors for new Web developers.

REVIEW QUESTIONS

1. What are the differences between Macintosh and Windows systems?
 - a. Macintoshes have smaller text and lower gamma
 - b. Macintoshes have larger text and lower gamma
 - c. Windows systems have smaller text and lower gamma
 - d. Windows systems have larger text and lower gamma
2. What is an average environment for most users?
 - a. Netscape on Macintosh with a 19" monitor
 - b. Internet Explorer on Windows with a 15" monitor
 - c. Netscape on Windows with a 17" monitor
 - d. Internet Explorer on Macintosh with a 13" monitor
3. What can you tell from analyzing your Web server log files?
 - a. How many of your users print your pages
 - b. How many of your users use Macs
 - c. How many of your users have 15" monitors
 - d. How many of your users view your pages in 16-bit color
4. What is the decimal equivalent of the hexadecimal value 10?
 - a. 8
 - b. 10
 - c. 16
 - d. A

5. What is the percentage of blue in the hexadecimal color #996633?
 - a. 80%
 - b. 60%
 - c. 40%
 - d. 20%
6. If the Photoshop Color Picker dialog box shows a red value of 204, a green value of 153, and a blue value of 0, what is the corresponding hexadecimal color?
 - a. #CF9600
 - b. #CC9900
 - c. #966900
 - d. #996600
7. What color is represented by the code #654321?
 - a. Pale green
 - b. Dark green
 - c. Orange
 - d. Dark red
8. What code below represents light blue?
 - a. #012345
 - b. #ABCDEF
 - c. #AAFF33
 - d. #FC9630
9. What happens when you use the word *green* to define a background color?
 - a. Nothing, because *green* must be capitalized
 - b. The background displays the hexadecimal color 008000, which is not a Web-safe color
 - c. The background displays the hexadecimal color 00FF00, which is not a Web-safe color
 - d. The background displays the hexadecimal color 00FF00, which is a Web-safe color
10. What tag is used primarily to lay out Web pages?
 - a. IMG
 - b. TABLE
 - c. BODY
 - d. DIV

11. Which parts of HTML must be capitalized?
 - a. Tags
 - b. Attributes
 - c. Values of the attributes
 - d. None of the above
12. Which attribute cannot be used with the IMG tag?
 - a. ALIGN
 - b. BORDER
 - c. BACKGROUND
 - d. HSPACE
13. Which tag uses the CELLSPACING attribute?
 - a. IMG
 - b. TABLE
 - c. TR
 - d. TD
14. What is the main use of Web-safe colors?
 - a. In images so they appear on both Macs and PCs
 - b. In HTML so pages appear on both Macs and PCs
 - c. In images so they appear on both Netscape and Internet Explorer
 - d. In HTML so pages appear on both Netscape and Internet Explorer
15. How do you use the LOWSRC tag?
 - a. Set the value to the same image.
 - b. Set the value to the same image, but use different HEIGHT and WIDTH attributes.
 - c. Set the value to a different image.
 - d. Don't set the value to anything; the LOWSRC attribute does not take a value.
16. Why would you set the value of the BORDER attribute in the IMG tag to 0?
 - a. To allow text to wrap
 - b. To remove horizontal margins
 - c. To remove both horizontal and vertical margins
 - d. To remove the colored outline

17. How do you make sure images can be reused in a page without forcing the browser to reload them from the server each time?
 - a. Put them in the cache.
 - b. Don't do anything special; the browser will put them in the cache.
 - c. Make sure the user clicks the Reload or Refresh button.
 - d. Duplicate the image each time you want to use it, and give it a different name.
18. You have two directories, one named `httpd` (containing your HTML files) and one named `other` (containing an image named `logo.gif`). Both directories are in the same parent directory. What is the file path needed to insert the logo into the HTML?
 - a. ``
 - b. ``
 - c. ``
 - d. ``
19. Which of the IMG tags successfully displays an image?
 - a. ``
 - b. ``
 - c. ``
 - d. `<SRC="Harold.GIF" ALT="Harold's birthday" HSPACE=2 BORDER=ON IMG>`
20. Which IMG tag below successfully displays an image?
 - a. ``
 - b. ``
 - c. ``
 - d. ``

HANDS-ON PROJECTS



Project 3-1 Prepare an Image for WebTV

Although less than 1% of your audience uses WebTV to view your site, your designer wants your site's logo to work on all platforms. Edit the image to look good when displayed on a television.

To edit an image for television in Photoshop:

1. In Photoshop, open **3-1.tif** from the Data Disk.

2. Change the Mode to **RGB color**.
3. To fix the colors, apply the **NTSC Colors** filter. This darkens the bright colors so they don't distort when shown on a television.
4. Fix the fine lines. Blur the image with the **Gaussian Blur** filter and set the Radius to **0.3**. This blurs the entire image slightly, preventing the fine lines from aliasing on the television. You also can use one of the selection tools to select the area around the lines and blur only that area.
5. Optimize and save the image as **3-1.gif** in the Chapter 3 folder on your hard drive.



Project 3-2 Make a Simple Web Page and Align an Image

You can complete this and the following exercises with a WYSIWYG HTML editor, but you are strongly encouraged to use a simple text editor such as SimpleText or NotePad to learn how pages are constructed.

To align an image on a Web page:

1. In Photoshop, open **3-2.tif** from the Data Disk.
2. Select the **Canvas Size** tool and record the height and width of the image.
3. Optimize and save the image as **fish.gif** in a new directory on your desktop called **project_3-2**.
4. Open a simple text editor and create a new file called **fish.html**. Save it to the same directory as the image.
5. Add an IMG tag to the HTML file which indicates the source of the image and includes the necessary attributes (SRC, HEIGHT, WIDTH, and ALT). Note that you do not need to add the HTML and BODY tags and that the order of the attributes does not matter. It should look like this:

```
<IMG SRC="fish.gif" HEIGHT="120" WIDTH="300" ALT="Fish">
```

6. Save the HTML file and view it in a browser.
7. Add about 50 words of your own above the IMG tag.
8. Add another 50 words of your own below the IMG tag.
9. Within the IMG tag, add these attributes: ALIGN set equal to **right**, HSPACE set equal to **10**, VSPACE set equal to **5**. It should look like this:

```
<IMG SRC="fish.gif" HEIGHT="120" WIDTH="300" ALT="Fish"
ALIGN="right" HSPACE="10" VSPACE="5">
```
10. Save the file and view it in a browser. The image should be aligned on the right, with a 10-pixel margin on the sides and a 5-pixel margin on the top and bottom.



Project 3-3 Use a Low-Source Image

The large fish image takes several seconds to load when using a slow modem. You want the rest of the page to load without having to wait for the image. Create a quickly loading version of the image, and include it using the LOWSRC attribute.

To use the LOWSRC attribute to quickly load a large image:

1. In Photoshop, open **fish.gif** from the Data Disk.
2. Set the Mode to **RGB Color**, and then set it to **Indexed Color**.
3. In the Indexed Color dialog box, set the palette to **4 colors**.
4. Save the image as **fish_lowsrc.gif** in the directory called project_3-2.
5. In a text editor, open the **fish.html** file you created in Project 3-2, and add the LOWSRC attribute to the IMG tag to reference the new image. The HTML should look like this:

```
<IMG SRC="fish.gif" HEIGHT="120" WIDTH="300" ALT="Fish"
ALIGN="right" HSPACE="10" VSPACE="5"
LOWSRC="fish_lowsrc.gif">
```
6. Save the file as **fish2.html** and view it in a browser. Unless you have a very fast computer, you should see the low-source image load and then be replaced by the final image.



Project 3-4 Use an Image as a Button

One common use for small images is to set them up as links to other pages, which effectively makes them buttons users can click to navigate a Web site.

To use an image as a button:

1. In Photoshop, open **3-2.tif** from the Data Disk.
2. Make sure the Mode is set to **RGB color** and reduce the size to **100 × 100** pixels.
3. Save the image as **fish_sm.gif** in the directory named project_3-2.
4. In a text editor, create a new HTML file, and save it as **index.html** in the project_3-2 folder.
5. Add an IMG tag to display the image.
6. Make the image a link to fish.html by adding an anchor tag. The final HTML should look like this:

```
<A HREF="fish.html"><IMG SRC="fish_sm.gif"></A>
```
7. Save the HTML file and view it in a browser. The image will have a colored border around it.
8. To remove the colored border, add the **BORDER** attribute to the IMG tag and set the value to **0**.
9. Save the HTML file in the project_3-2 folder.



Project 3-5 Set File Paths to Display Images in Other Directories

A common mistake for new Web developers is to set file paths incorrectly. Practice using paths by referencing images in different directories.

To set a file path in HTML:

1. On your desktop, create a directory named **project_3-5**.
2. Within the directory, create two more directories. Name one **html** and name the other **images**.
3. Copy the files from Projects 3-1 through 3-4 to these new directories. Put **index.html** and **fish.html** into the **html** directory, and put **fish.gif** and **fish_sm.gif** into the **images** directory.
4. In a text editor, edit **index.html** to reference the image from the new directory. Set the SRC attribute in the IMG tag to **"../images/fish_sm.gif"**. The IMG tag should look like this:

```
<A HREF="fish.html"><IMG SRC="../images/fish_sm.gif"
BORDER=0></A>
```
5. Edit **fish.html** in the same way, changing the values of the SRC attribute.
6. View **index.html** in a browser, and click the **image** to open **fish.html**. The images should appear on both pages.



Project 3-6 Place an Image in a Page with a Colored Background

Many Web pages use colored backgrounds. To make the image fit in, it should not have a different background color. Create a page with a colored background, and include an image with the same background color.

To insert an image with a colored background on a Web page with a colored background:

1. In a text editor, create a new HTML file named **index.html** and save it in a new directory called **project_3-6**.
2. Give the page a pale turquoise background by using the BACKGROUND attribute of the BODY tag. Add this tag to the file: **<BODY BGCOLOR="#99FFCC">**.
3. A few lines below, add the closing tag: **</BODY>**.
4. Open file **3-6.tif** from the Data Disk and save it as **car.gif** in the **project_3-6** directory. Leave the file open in Photoshop.
5. In the HTML file add an **IMG** tag between the BODY tags. Set the **SRC** attribute to **car.gif**.
6. Save the file and view it in a browser. You will see a white background around the image of the car.

7. Set the background color of the image to the same color as the page. In Photoshop, open the Color Picker dialog box for the foreground color and set the color to **99FFCC** in the hexadecimal box near the bottom. If you are using an older version which does not allow you to enter hexadecimal color, set red to **153**, green to **255**, and blue to **204**.
8. Select the **Paint Bucket** tool and look at the Options palette. Set the tolerance to **30** and make sure the **Anti-aliased** and **Contiguous** check boxes are selected. If you do not do this, you will leave a white halo around the car.
9. Fill in the white background with the pale green using the Paint Can tool.
10. Save the **image**, then reload the file in the browser by clicking the **Reload** or **Refresh** button. The background of the image should match the color of the page.



Project 3-7 Create a Transparent Image

You might not know the background color of the Web page that will contain your image, or the image might have to appear on pages with different background colors. In these circumstances, make the image background transparent so that the image can appear on any background color. Create an image with a transparent background and include it in an HTML table with a background color.

To create an image with a transparent background:

1. In Photoshop, open **3-7.tif** from the Data Disk.
2. Convert the background layer to a normal layer. (Double-click the **background layer** in the layers palette, and then click **ok**.)
3. Select the **Magic Wand** tool. Set the tolerance to **32** and check the **Contiguous** box and the **Anti-aliased** box. Select the **white background** with the Magic Wand tool and then press the **Delete** key.
4. Change the Mode to **Indexed Color** and set the number of colors to **32**.
5. Make sure the Transparency box is selected and, for the Matte color, select the **pale turquoise** color. Set dither to none.
6. Click **OK** and save the image as **trans.gif** in a new directory named **project_3-7**.
7. In a text editor, create a new HTML file named **index.html** in the project folder (project_3-7).
8. Create a one-cell table with a pale turquoise background. The HTML should look like this:

```
<TABLE><TR><TD BGCOLOR="#99FFCC">
</TD></TR></TABLE>
```
9. Between the table delimiter tags, add the HTML code to reference the new image.
10. Save the **index.html** file to the directory called project_3-7 and view it in a browser. You should see a blue box containing the image, which now appears to have a pale turquoise background.



Project 3-8 Use Background Colors to Create a Smaller Image

Your designer has given you an image with large colored margins. You know that you can use background colors in HTML instead of creating large margins. The only trick is matching the HTML color to the image color. The image is a high-color JPEG image, and you cannot make the background transparent. Crop the image and place it in a table with the same color for the background.

To crop an image and include it in a table with a colored background:

1. In Photoshop, open **3-8.tif** from the Data Disk.
2. Use the **Eyedropper** tool to select the color of the background.
3. Look at the Foreground Color in the Color Picker dialog box to see its hexadecimal code. The color should be Web-safe so that the image matches the table color. HTML colors are three pairs of numbers; to be Web-safe, each pair must be 00, 33, 66, 99, CC, or FF. The color looks Web-safe at first glance, but it isn't. You can tell because the color cube icon appears near the top of the window. Click the **color cube** icon to make the color Web-safe. The color shifts to the nearest Web-safe color. Record the hexadecimal code.
4. Select the **Paint Bucket** tool, set the options to a tolerance of **30**, and select **Anti-aliasing**.
5. Replace the background color with the new Web-safe version.
6. Open the **Canvas Size** dialog box to see the image dimensions. Record the height and width.
7. Trim the image to remove all but a few pixels of the background color.
8. Optimize the image and save it as **3-8.jpg** in a new project directory named **project_3-8**. Use a high-quality setting.
9. In a text editor, create a new HTML file named **index.html**, and save it in the same directory.
10. Create a table of the same dimensions as the original image. Make the background color the same as the new Web-safe version of the image's background color. The HTML code should look like this:

```
<TABLE HEIGHT="120" WIDTH="160" BGCOLOR="6699CC"
BORDER="0"><TR><TD ALIGN="CENTER">
</TD></TR></TABLE>
```
11. Between the table delimiter tags, add an **image tag** to reference the new image.
12. Save the **file** and view it in a browser. What you see should look like the original image.

CASE PROJECT



By now, you should have collected several images to use in your online portfolio project, and optimized them for the Web. Your next step is to create Web pages to display the images. Using a WYSIWYG editor or a simple text editor, create a biography page that includes a picture of you. Create a gallery page that contains your images. The pages should link to each other. Preview the pages on at least two sizes of monitors, on both Netscape and IE, and both Macintosh and Windows. Notice the differences in how the pages appear. Edit the images and the Web pages to compensate for the differences.